UNITED STATES PATENT AND TRADEMARK OFFICE

UNITED STATES DEPARTMENT OF COMMERCE
United States Patent and Trademark Office
Address: COMMISSIONER FOR PATENTS
P.O. Box 1450
Alexandria, Virginia 22313-1450
www.uspto.gov

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/039,289	01/04/2002	Gilbert Wolrich	10559-612001/P12851	8345

20985 7590 09/28/2005

FISH & RICHARDSON, PC
12390 EL CAMINO REAL
SAN DIEGO, CA 92130-2081

EXAMINER	
CHANNAVAJJALA, SRIRAMA T	
ART UNIT	PAPER NUMBER
2166	

DATE MAILED: 09/28/2005

Please find below and/or attached an Office communication concerning this application or proceeding.

Office Action Summary	Application No.	Applicant(s)	
	10/039,289	WOLRICH ET AL.	
	Examiner	Art Unit	
	Srirama Channavajjala	2166	

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE 3 MONTH(S) FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If the period for reply specified above is less than thirty (30) days, a reply within the statutory minimum of thirty (30) days will be considered timely.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *01 August 2005*.

2a)☐ This action is **FINAL**. 2b)☒ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-37* is/are pending in the application.

4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-37* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

a)☐ All b)☐ Some * c)☐ None of:

1.☐ Certified copies of the priority documents have been received.

2.☐ Certified copies of the priority documents have been received in Application No. _____.

3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1) ☒ Notice of References Cited (PTO-892)
2) ☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)
3) ☐ Information Disclosure Statement(s) (PTO-1449 or PTO/SB/08) Paper No(s)/Mail Date _____.
4) ☐ Interview Summary (PTO-413) Paper No(s)/Mail Date. _____ .
5) ☐ Notice of Informal Patent Application (PTO-152)
6) ☐ Other: _____.

DETAILED ACTION

Response to Office action mailed on May 17,2005

1. In view of applicant's remarks at page 2-3, claims 1-37, examiner hereby issuing ***"non-final office action"***

2. Claims 1-37 are pending in this application.

3. Examiner acknowledges applicant's "*terminal disclaimer*" under 37 CFR 3.73(b) and 1.32(b) filed on June 15, 2005.

4. A request for continued examination under 37 CFR 1.114, including the fee set forth in 37 CFR 1.17(e), was filed in this application after final rejection. Since this application is eligible for continued examination under 37 CFR 1.114, and the fee set forth in 37 CFR 1.17(e) has been timely paid, the finality of the previous Office action has been withdrawn pursuant to 37 CFR 1.114. Applicant's submission filed on 18 April 2005 has been entered and a non-final Office action, mailed on May 17, 2005.

5. Claims 1-7, 14, 26-28 have been amended [4/18/2005].

6. Claims 31-37 have been added [4/18/2005].

7. Examiner acknowledges applicant's amendment filed on 8/12/2004.

8. In view of applicant's amendment to claim 20, the objection to the claim 20 as set forth in the previous office action is hereby withdrawn.

9. In view of applicant's submitted arguments [see page 10-11], the rejection of claims 1-25 under 35 USC 101 set forth in the previous office action is hereby withdrawn

10. In view of applicant's filed "Terminal Disclaimer under 37 CFR 3.73(b) and 1.32(b), the non statutory double patenting rejection set forth in the previous office action is hereby withdrawn.

Drawings

11. Examiner acknowledges applicant filed drawings are acceptable for examination, and in view of applicant's amendment to the figs 2-4, the objection to the drawings set forth in the previous office action is hereby withdrawn.

Information Disclosure Statement

12. The electronic information disclosure statement (eIDS) submitted on 28 June 2004 is in compliance with the provisions of 37 CFR 1.97. Accordingly, the information disclosure statement is being considered by the examiner, a copy of electronic information disclosure statement [1-7 pages] enclosed with previous office action.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

13. Claims 1-37 are rejected under 35 U.S.C. 103(a) as being unpatentable over Regache, Pascal [hereafter Pascal], EP 0760501 A published on 05 March 1997 in view of Slane US Patent No. 6438651.

14. As to claim 1,26,31,36, Pascal teaches a system which including 'storing in memory a plurality of a queue descriptors [col 5, line 14-21,line 22-33], Pascal specifically teaches memory having multiple memory pages that corresponds to storing in memory, further Pascal also teaches number of data items storage locations that corresponds to memory pages and data item storage locations in a circular queue, queue status that corresponds to queue descriptors,; 'each including a head pointer pointing to a first element in a corresponding queue and a tail pointer pointing to a last element in the corresponding queue' [col 7, line 14-20, fig 2-3], Pascal specifically teaches circular queue maintenance of head and tail pointers, head pointer corresponds to fig 3, element 32; tail pointer corresponds to fig 3, element 30;

'fetching from memory to a cache one of either the head pointer or tail pointer of a first queue descriptor corresponding to the first queue' [col 9, line 1-15], Pascal

specifically teaches checking the queue status, further data item storage locations indicated by tail pointer held in register as detailed in fig 7;

'returning to the memory from the cache portions of the first queue descriptor modified by the operation' [col 9, line 35-38];

'a count identifying a number of elements in the queue' [col 6, line 10-19, col 8, line 4-12 as claim in claim 31, claim 36], Pascal specifically teaches maintaining a count indicative of the number of data items storage locations, also to distinguish current segments by using flags, further Pascal also teaches queue status information for example P-index is incremented each time a data item is written to the queue, therefore, Pascal teaches count that identifying queue size and status as detailed in col 8, line 4-12].

It is however noted that Pascal does not specifically teach 'in response to a command to perform an enqueue or dequeue operation with respect to a first queue', although Pascal specifically teaches checking the queue status, determining the data item storage locations in particular segments as detailed in fig 3. On the other hand, Slane specifically teaches 'in response to a command to perform an enqueue or dequeue operation with respect to a first queue' [col 3, line 52-61, fig 2], Slane specifically teaches enqueue, dequeue operations with respect to queue as detailed in fig 2, enqueue, dequeue operations corresponds to Slane's enqueue, dequeue as shown in fig 2.

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because both Slane, Pascal are directed to queue operations with respect to memory data, more specifically, Pascal is directed to entity for writing data items in sequence to a circular queue formed in paged memory from N data-item storage locations [see col 3, line 3-8], while Slane is directed to managing requests to a cache using flags to queue and dequeue data in a buffer, more specifically, cache management scheme that optimizes read and write hits for queues, each subsequent read/dequeue and write/enqueue operations request to the circular buffer queue as detailed in col 2, line 38-45] and are from same field of endeavor.

One of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because that would have allowed users of Pascal to use enqueue dequeue operations with respect to queue not only optimizes access requests during subsequent access requests to the queue, but also read and write are assured as each subsequent read/dequeue and write/enqueue request as suggested by Slane [col 2, line 52-59] bringing the advantages of improving queue performance and accessing to the data.

15. As to Claim 2, 15, both Pascal and Slane disclosed 'fetching the head pointer and not the tail pointer of the first queue descriptor in response to a command to perform a dequeue operation with respect to the first queue' [Pascal: col 7, line 55-9, col 8, line 1, col 9, line 11-15; Slane: col 3, line 52-55, col 4, line 47-49, col 5, line 61-63, fig 3].

16. As to Claim 3, 16, Slane disclosed 'fetching the tail pointer and not the head pointer of the first queue descriptor in response to a command to perform an enqueue operation with respect to the first queue' [col 4, line 47-51, fig 3].

17. As to Claim 4, 17, Slane disclosed 'returning to memory the head pointer and not the tail pointer of the first queue descriptor if only dequeue operations were performed on the first queue' [col 4, line 51-52, line 56-60, fig 4-6].

18. As to Claim 5, 18, Slane disclosed 'returning to memory the tail pointer and not the head pointer of the first queue descriptor if only enqueue operations were performed on the first queue while the first queue was unempty' [col 5, line 64-67, col 6, line 1-8].

19. As to Claim 6, 19, Slane disclosed 'returning to memory the head pointer and tail pointer of the first queue descriptor if an enqueue and a dequeue operation were performed on the first queue, or an enqueue operation was performed on the queue while the first queue was empty' [col 7, line 3-8].

20. As to claim 7, 33, Pascal teaches a system which including 'storing in memory a plurality of a queue descriptors [col 5, line 14-21,line 22-33], Pascal specifically teaches memory having multiple memory pages that corresponds to storing in memory, further Pascal also teaches number of data items storage locations that corresponds to memory pages and data item storage locations in a circular queue; queue status that corresponds to queue descriptors;

'each including a head pointer pointing to a first element in a corresponding queue and a tail pointer pointing to a last element in the corresponding queue' [col 7, line 14-20, fig 2-3], Pascal specifically teaches circular queue maintenance of head and tail pointers, head pointer corresponds to fig 3, element 32; tail pointer corresponds to fig 3, element 30;

'determining whether a head pointer or a tail pointer of a queue descriptor that was fetched from the memory to a cache' [fig 3, col 7, line 14-16], head pointer corresponds to fig 3, head pointer, element 32; tail pointer corresponds to fig 3, element 30;

'returning one of either the head pointer or tail pointer to the memory from the cache only if that pointer had been modified' [col 9, line 35-38],

'a count identifying a number of elements in the queue' [col 6, line 10-19, col 8, line 4-12], Pascal specifically teaches maintaining a count indicative of the number of data items storage locations, also to distinguish current segments by using flags, further Pascal also teaches queue status information for example P-index is incremented each

time a data item is written to the queue, therefore, Pascal teaches count that identifying queue size and status as detailed in col 8, line 4-12].

It is however noted that Pascal does not specifically teach 'in response to an enqueue or dequeue operation had been modified by the enqueue or dequeue operation', although Pascal specifically teaches checking the queue status, determining the data item storage locations in particular segments as detailed in fig 3. On the other hand, Slane specifically teaches "in response to an enqueue or dequeue operation had been modified by the enqueue or dequeue operation' [col 3, line 52-61, fig 2], Slane specifically teaches enqueue, dequeue operations with respect to queue as detailed in fig 2, enqueue, dequeue operations corresponds to Slane's enqueue, dequeue as shown in fig 2.

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because both Slane, Pascal are directed to queue operations with respect to memory data, more specifically, Pascal is directed to entity for writing data items in sequence to a circular queue formed in paged memory fron N data-item storage locations [see col 3, line 3-8], while Slane is directed to managing requests to a cache using flags to queue and dequeue data in a buffer, more specifically, cache management scheme that optimizes read and write hits for queues, each subsequent read/dequeue and write/enqueue operations request to the circular buffer queue as detailed in col 2, line 38-45] and are from same field of endeavor.

One of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because that would have allowed users of Pascal to use enqueue dequeue operations with respect to queue not only optimizes access requests during subsequent access requests to the queue, but also read and write are assured as each subsequent read/dequeue and write/enqueue request as suggested by Slane [col 2, line 52-59] bringing the advantages of improving queue performance and accessing to the data.

21. As to Claim 8, 20, Slane disclosed 'using valid bits in the cache to track modifications to the pointers' [col 6, line 19-25].

22. As to Claim 9, 21, Slane disclosed 'using a first valid bit to track modifications to the head pointer and second valid bit to track modifications to the tail pointer' [col 6, line 26-37].

23. As to Claim 10, 22, 29, Slane disclosed 'setting the first valid bit if a dequeue operation is performed with respect to the queue descriptor' [col 6, line 57-64], 'an enqueue operations performed with respect to the queue descriptor while the queue is empty' [col 7, line 3-12].

24. As to Claim 11, 23, Slane disclosed 'setting the second valid bit if an enqueue operation is performed with respect to the queue descriptor' [col 4, line 65-67, col 5, line 1-8].

25. As to Claim 12, 24, Slane disclosed 'setting a pointer's valid bit when the pointer is fetched from the memory to the cache' [col 5, line 55-63].

26. As to Claim 13, 25, 30, Slane disclosed 'returning to the memory pointers whose valid bits have been set' [col 5, line 59-63].

27. As to claim 14,34, Pascal teaches a system which including 'memory for storing queue descriptors [col 5, line 14-21,line 22-33], Pascal specifically teaches memory having multiple memory pages that corresponds to storing in memory, further Pascal also teaches number of data items storage locations that corresponds to memory pages and data item storage locations in a circular queue, queue status that corresponds to queue descriptors

'a count identifying a number of elements in the queue' [col 6, line 10-19, col 8, line 4-12 as in claim 34], Pascal specifically teaches maintaining a count indicative of the number of data items storage locations, also to distinguish current segments by using flags, further Pascal also teaches queue status information for example P-index is incremented each time a data item is written to the queue;

'each of which include a head pointer pointing to a first element in a corresponding queue and a tail pointer pointing to a last element in the corresponding queue' [col 7, line 14-20, fig 2-3], Pascal specifically teaches circular queue maintenance of head and tail pointers, head pointer corresponds to fig 3, element 32; tail pointer corresponds to fig 3, element 30;

'a cache for storing queue descriptors corresponding to up to a number of the memory's queue descriptors' [col 8, line 26-31];

'a processor configured to: fetching from memory to a cache one of either the head pointer or tail pointer of a first queue descriptor corresponding to the first queue' [col 9, line 1-15], Pascal specifically teaches checking the queue status, further data item storage locations indicated by tail pointer held in register as detailed in fig 7;

'return to the memory from the cache portions of the first queue descriptor modified by the operation' [col 9, line 35-38], It is however noted that Pascal does not specifically teach 'in response to a command to perform an enqueue or dequeue operation with respect to the particular queue descriptor', although Pascal specifically teaches checking the queue status, determining the data item storage locations in particular segments as detailed in fig 3. On the other hand, Slane specifically teaches 'in response to a command to perform an enqueue or dequeue operation with respect to the particular queue descriptor' [col 3, line 52-61, fig 2], Slane specifically teaches enqueue, dequeue operations with respect to queue as detailed in fig 2, enqueue, dequeue operations corresponds to Slane's enqueue, dequeue as shown in fig 2.

It would have been obvious to one of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because both Slane, Pascal are directed to queue operations with respect to memory data, more specifically, Pascal is directed to entity for writing data items in sequence to a circular queue formed in paged memory from N data-item storage locations [see col 3, line 3-8], while Slane is directed to managing requests to a cache using flags to queue and dequeue data in a buffer, more specifically, cache management scheme that optimizes read and write hits for queues, each subsequent read/dequeue and write/enqueue operations request to the circular buffer queue as detailed in col 2, line 38-45] and are from same field of endeavor.

One of the ordinary skill in the art at the time of applicant's invention to incorporate the teachings of Slane into data handling system with circular queue formed in paged memory of Pascal because that would have allowed users of Pascal to use enqueue dequeue operations with respect to queue not only optimizes access requests during subsequent access requests to the queue, but also read and write are assured as each subsequent read/dequeue and write/enqueue request as suggested by Slane [col 2, line 52-59] bringing the advantages of improving queue performance and accessing to the data.

28. As to claim 27, 32, 35, 37, Slane disclosed 'fetch the head pointer and not the tail pointer of the first queue descriptor in response to a command to perform a dequeu operation with respect to the first queue' [col 3, line 52-55, col 4, line 47-49, col 5, line 61-63, fig 3];

'fetching the tail pointer and not the head pointer of the first queue descriptor in response to a command to perform an enqueue operation with respect to the first queue" [col 4, line 47-51, fig 3].

29. As to Claim 28, Slane disclosed 'the head pointer and not the tail pointer of the first queue descriptor if only dequeue operations are performed on the first queue'[col 4, line 51-52, line 56-60, fig 4-6],

'the tail pointer and not the head pointer of the first queue descriptor if only enqueue operations are performed on the queue while the first queue was unempty' [col 5, line 64-67, col 6, line 1-8],

'both the head pointer and tail pointer of the first queue descriptor if both an enqueue and a dequeue are performed on the queue, or an enqueue operation was performed on the queue while the first queue was empty' [col 7, line 3-8].

Response to Arguments

30. Applicant's arguments filed on 8/1/2005, claims 1-37 have been fully considered, for examiner's response see the discussion below:

Examiner recognized the typo error at page 4, that Pascal teaches a system that includes "storing in memory a plurality of queue descriptors" should be col 5, 33-39 and Pascal teaches claims 31,33,34,36 limitation "a count identifying a number of elements in the queue" [col 6, line 10-19, col 8, line 4-12], Pascal specifically teaches maintaining a count indicative of the number of data items storage locations, also to distinguish current segments by using flags, further Pascal also teaches queue status information for example P-index is incremented each time a data item is written to the queue as noted in the above office action;

a) At page 2, claim 1-30, applicant argues that "nowhere does Pascal, alone or in combination with Slane, teach or suggest even using a plurality of queues, much less, storing in memory.....

As to the argument [a], as best understood by the examiner, Pascal is directed to circular queue formed in paged memory, more specifically, entity for writing reading data items in sequence in a circular queue formed in page memory having "N" data item storage locations [col 3, line 3-8], it is also noted that Pascal specifically teaches memory having multiple memory pages that corresponds to storing in memory, further Pascal also teaches number of data items storage locations that corresponds to

memory pages and data item storage locations in a circular queue, and further queue status that corresponds to queue descriptors [col 5, line 14-21,line 22-33].

b) At page 3, claims 1-30, applicant argues that "none of these is described as a plurality of items each including a head pointer and a tail pointer".

As to the above argument [b], as best understood by the examiner, Pascal specifically teaches circular queue itself maintains head and tail pointers, further these pointers are maintained within a queue segment or storage locations as detailed in col 7, line 11-22, fig 3, therefore, Pascal teaches both head and tail pointers.

c) At page 3, claims 31,33,34,36, applicant argues that Pascal and Slane neither describes nor suggests at least "a count identifying a number of elements in the queue", examiner has not even argued that the references teach or suggest this limitation.

As to the above argument, in the above office action, examiner noted that Pascal teaches 'a count identifying a number of elements in the queue' [col 6, line 10-19, col 8, line 4-12], Pascal specifically teaches maintaining a count indicative of the number of data items storage locations, also to distinguish current segments by using flags, further Pascal also teaches queue status information for example P-index is incremented each time a data item is written to the queue, therefore, Pascal teaches count that identifying queue size and status as detailed in col 8, line 4-12].

Conclusion

The prior art made of record

a. US Patent No. 6438651

b. EP0760501 A

Any inquiry concerning this communication or earlier communications from the examiner should be directed to Srirama Channavajjala whose telephone number is 571-272-4108. The examiner can normally be reached on Monday-Friday from 8:00 AM to 5:30 PM Eastern Time.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, Alam, Hosain, T, can be reached on (571) 272-3978. The fax phone numbers for the organization where the application or proceeding is assigned is 703/872-9306 Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free)

sc
Patent Examiner.
September 26, 2005.

SRIRAMA CHANNAVAJJALA
PRIMARY EXAMINER